UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,537	03/28/2001	Randall K. Curey	P573C	6434

23586 7590 03/26/2004

ROBERT E MALM
16624 PEQUENO PLACE
PACIFIC PALISADES, CA 90272

EXAMINER	
MOSLEHI, FARHOOD	
ART UNIT	**PAPER NUMBER**
2154	4

DATE MAILED: 03/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/821,537	Applicant(s) CUREY ET AL.	
	Examiner Farhood Moslehi	Art Unit 2154	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-49 are presented for examination.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claim 1,2,6-9,21-22,25-27,31-34,46,47 is rejected under 35 U.S.C. 102(b) as being anticipated by Maitra (5,623,647).

4. As per claim 1, Maitra discusses a method for repetitively executing a plurality of software packages at one or more rates, utilizing a common set of computational resources, the method comprising the steps:

Generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages (e.g. col. 6, lines 34-39); executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals (e.g. col. 6, lines 40-55).

5. As per claim 25, it is rejected for the similar reasons as stated above.

6. As per claim 26, it is rejected for the similar reasons as stated above.

7. As per claim 27, it is rejected for the similar reasons as stated above.

8. As per claim 2, Maitra teaches the method wherein the plurality of software packages of the "executing" step includes only valid software packages, the method further comprising the step:

Utilizing one or more tests to identify the software packages that are valid (e.g. col.6, lines 56-67).

9. As per claim 6, Maitra teaches a method wherein a software package is assigned its own dedicated memory region, the software package's dedicated memory region including a stock memory region and /or a heap memory region, one of the tests for validity being whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region (e.g. col. 7, lines 66-67 and col. 8, lines 1-20).

10. As per claim 31, it is rejected for the similar reasons as stated above.

11. As per claim 8, it is rejected for the similar reasons as stated above.

12. As per claim 33, it is rejected for the similar reasons as stated above.

13. As per claim 9, it is rejected for the similar reasons as stated above.

14. As per claim 34, it is rejected for the similar reasons as stated above.

15. As per claim 7, Maitra teaches a method wherein one of the tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time (e.g. col. 3, lines 4-19).

16. As per claim 32, it is rejected for the similar reasons as stated above.

17. As per claim 21, Maitra shows a method wherein an executive software package enforces the discipline that each software package executes only during the time intervals of its sequence of time intervals, the executive software package determining when the execution of a software package extends into a time interval belonging to the sequence of time intervals assigned to another software package and performs a remedial action (e.g. col. 4, lines 22-35). The Operating System provides the above functionality for any number of software packages or applications.

18. As per claim 46, it is rejected for the similar reasons as stated above.

19. As per claim 22, Maitra shows a method wherein the presence of those software packages that are present is detected (e.g. col. 4, lines 22-35). It is inherent for any Operating System to detect the presence of applications that are either queued or presently in execution.

20. As per claim 47, it is rejected for the similar reasons as stated above.

Claim Rejections - 35 USC § 103

21. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. Claims 3-5,18-20,28-30,43-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Jablon et al (5,421,006) (hereinafter Jablon).

23. As per claim 3, Maitra does not discuss the method of the above claim wherein one of the tests for validity is a one's complement checksum test of a software package's program memory. Jablon teaches the method of the above claim wherein one of the tests for validity is a one's complement checksum test of a software package's program memory (e.g. col. 5, lines 4-18). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to test the validity of the software application using a one's compliment checksum method.

24. As per claim 28, it is rejected for the similar reasons as stated above.

25. As per claim 4, Maitra does not teach a method wherein a software package is assigned its own dedicated memory region, one of the tests for validity being whether the address retuned for the software package's initialization procedure lies within its dedicated memory region. Jablon teaches a method wherein a software package is assigned its own dedicated memory region, one of the tests for validity being whether the address retuned for the software package's initialization procedure lies within its dedicated memory region (e.g. col. 6, lines 27-37). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to have set memory partitions for each software package and the validity test being conducted within predefined memory regions.

26. As per claim 29, it is rejected for the similar reasons as stated above.

27. As per claim 5, Maitra does not discuss a method wherein one of the tests is whether the address is returned within a predetermined time. Jablon teaches a method

wherein one of the tests is whether the address is returned within a predetermined time (e.g. col. 7, lines 21-34). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to have a predetermined time to test the software validity.

28. As per claim 30, it is rejected for the similar reasons as stated above.

29. As per claim 18, Maitra does not show a method wherein safety-critical software is places in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software. Jablon shows a method wherein safety-critical software is places in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software (e.g. col. 6, lines 27-38). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to isolate different software packages in different partitions.

30. As per claim 43, it is rejected for the similar reasons as stated above.

31. As per claim 19, Maitra does not teach a method wherein each of the plurality of software packages is assigned its own memory block, a software package being enabled to read data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks readable by a software package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules. Jablon teaches a method wherein each of the plurality of software packages is assigned its own memory block, a software package being enabled to read data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks readable by a software

package being either predetermined or determined during execution of the software packages in accordance with a set of one or more rules (e.g. col. 8, lines 16-33). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to provide for software packages to read and write memory blocks from other software packages in accordance with a predetermined set of rules.

32. As per claim 20, it is rejected for the similar reason as stated above.

33. As per claim 44, it is rejected for the similar reasons as stated above.

34. As per claim 45, it is rejected for the similar reasons as stated above.

35. Claims 10,12,35 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Reznak (6,223,201).

36. As per claim 10, Maitra does not teach a method wherein a software package includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed. Reznak teaches a method wherein a software package includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed (e.g. col. 1, lines 25-32). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Reznak in order to have a background and a foreground application processing whereby the background process executes when the foreground processes cease to operate.

37. As per claim 12, Maitra teaches the method wherein the software package causes the power utilized in executing the software package to be minimized after completion of the background task (e.g. col. 2, lines 39-48).

38. As per claim 35, it is rejected for the similar reasons as stated above.

39. As per claim 37, it is rejected for the similar reasons as stated above.

40. Claims 11 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Reznak and further in view of Davidson et al. (6,292,934) (hereinafter Davidson).

41. As per claim 11, Maitra does not show the method wherein a background task is an infinite loop. Davidson teaches the method wherein a background task is an infinite loop (e.g. col. 17, lines 32-58). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Davidson in order to have the background process run in an infinite loop.

42. As per claim 36, it is rejected for the similar reasons as stated above.

43. Claims 13,14,38 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Yen (6,381,694).

44. As per claim 13, Maitra does not specifically teach a method wherein a failure in the execution of a software package causes information to be logged in a failure log. Yen teaches a method wherein a failure in the execution of a software package causes information to be logged in a failure log (e.g. col. 6, lines 49-59). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Yen in order to log the failure of a software package.

45. As per claim 38, it is rejected for the similar reasons as stated above.

46. As per claim 14, Maitra does not show a method wherein a failure in execution is linked to the software package that caused the failure. Yen shows a method wherein a failure in execution is linked to the software package that caused the failure (e.g. col. 5, lines 1-16 and Figures 3 and 9). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Yen in order to observe whether or not the failure is caused by a software package.

47. As per claim 39, it is rejected for the similar reasons as stated above.

48. Claims 15-17 and 40-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Yen and in further view of Harper et al. (6,629,266) (hereinafter Harper).

49. As per claim 15, Maitra does not specifically discuss the method wherein quality of performance in executing a software package is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in a failure log, the execution of a software package being subject to a plurality of execution options, an execution option being selected on the basis of one or more performance-quality parameter values. Harper teaches the method wherein quality of performance in executing a software package is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in a failure log, the execution of a software package being subject to a plurality of execution options, an execution option being selected on the basis of one

or more performance-quality parameter values (e.g. col. 13, lines 35-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Harper in order to provide for a performance based system whereby the log file can monitor and determine the packages performance.

50. As per claim 40, it is rejected for the similar reasons as stated above.

51. As per claim 16, Maitra does not teach a method wherein the plurality of execution options are user configurable. Harper teaches a method wherein the plurality of execution options are user configurable (e.g. Figure 5). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Harper in order to provide for user configurable parameters.

52. As per claim 41, it is rejected for the similar reasons as stated above.

53. As per claim 17, Maitra does not teach a method wherein performance-quality parameters include the number pf failures and/or the rate of failures for one or more classes of failures recorded in a software package's failure log. Harper shows a method wherein performance-quality parameters include the number pf failures and/or the rate of failures for one or more classes of failures recorded in a software package's failure log (e.g. col. 2, lines 32-35 and Figure 1). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Harper in order to provide for performance monitoring based on failure rate.

54. As per claim 42, it is rejected for the similar reasons as stated above.

55. Claim 23 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Potter et al. (5,014,327) (hereinafter Potter).

56. As per claim 23, Maitra does not describe a method wherein one or more of the plurality of software packages are independently compiled, linked and loaded. Potter shows a method wherein one or more of the plurality of software packages are independently compiled, linked and loaded (e.g. col. 7, lines 40-57). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Potter to provide for a software package to be independently complied, linked and loaded by the Operating System.

57. As per claim 48, it is rejected for the similar reasons as stated above.

58. Claims 24 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maitra in view of Circello et al. (5,761,491) (hereinafter Circello).

59. As per claim 24, Maitra does not show a method wherein a software package has its own stack, the software package's stack being selected prior to executing the software package. Circello shows a method wherein a software package has its own stack, the software package's stack being selected prior to executing the software package (e.g. col. 35-51). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Circello in order for the software's stack to be executed and selected by a process.

As per claim 49, it is rejected for the similar reasons as stated above.

Response to Amendment

60. Applicant's arguments filed on 1/22/2004 have been fully considered but are not persuasive.

61. In the remarks, applicants argued in substance that (1) Maitra does not disclose repetitively executing a plurality of software packages at one or more rates.

62. As to point (1) the examiner disagrees because Maitra discusses changing the clock speed of the processor running two applications (e.g. col. 6, lines 34-42).

63. In the remarks, applicants argued in substance that (2) Maitra does not disclose generating a sequence of non-overlapping time intervals for each of the plurality of software packages.

64. As to point (2) the examiner disagrees because Maitra discusses a multi-tasking environment whereby several applications are run simultaneously. The operation of each software would not affect the operation of other software programs running at same or different speeds (e.g. col. 6, lines 20-34).

65. In the remarks, applicants argued in substance that (3) Maitra does not disclose executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals.

66. As to point (3) the examiner disagrees because Maitra states that the first application is executed in its time quantum and the second application is executed in its time quantum, but at a different clock speed. Each application has a specific time quantum for being executed albeit at a different clock speed (e.g. col. 6, lines 39-43).

67. In the remarks applicants argued in substance that (4) Maitra does not require that the plurality of software packages executed by the "executing" means includes only valid software packages.

68. As to point (4) the examiner disagrees because Maitra states that the clock scheduling unit determines the computing requirements of the application (e.g. col. 8, lines 60-64). An invalid application would not be scheduled and no computing resources would be allocated.

69. In the remarks applicants argued in substance that (5) Maitra does not disclose a means for utilizing one or more tests to identify the software packages that are valid.

70. As to point (5) the examiner disagrees because Maitra discusses the computing requirements of an application (e.g. block 630 of figure 6, and col. 8, lines 60-64). A non-valid application would not have its configuration information being read into the memory.

71. In the remarks, applicants argued in substance that (6) Maitra does not require the plurality of software packages executed by the "executing" step includes only valid software packages.

72. As to point (6) the examiner disagrees because Maitra discusses the computing requirements of each application being extracted by the application characterization unit 220. A non-valid application would not be have an associated configuration file (e.g. col. 6, lines 59-63)

73. In the remarks, applicants argued in substance that (7) Maitra does not disclose utilizing one or more tests to identify the software packages that are valid.

74. As to point (7) the examiner disagrees because Maitra discusses the requirements for a an application computing requirements by running a benchmark evaluator would be a test to determine an application validity (e.g. col. 6, lines 62-67).

75. In the remarks, applicants argued in substance that (8) Maitra says nothing about these three sets of executable instructions being stored in dedicated regions of memory.

76. As to point (8) the examiner disagrees because Maitra discusses memory 420 can store instructions or code that are part of application programs (e.g. col. 8, lines 1-4). Since these applications programs can run at different clock cycles, hence different speeds, they would be stored in their own dedicated memory regions.

77. In the remarks, applicants argued in substance that (9) maitra does not say anything about a software package's dedicated memory region including a stack memory region and/or heap memory region.

78. As to point (9) the examiner disagrees because Maitra discusses memory 420 can store instructions or code that are part of application programs (e.g. col. 8, lines 1-4). Since these applications programs can run at different clock cycles, hence different speeds, they would be stored in their own dedicated memory regions.

79. In the remarks, applicants argued that (10) Maitra does not discuss whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region.

80. As to point (10) the examiner disagrees because Maitra shows that the application characterization unit 320, determines the computing requirements of an application (e.g. col. 7, lines 4-17).

81. In the remarks, applicants argued that (11) Maitra does not say anything about these three sets of executable instructions being stored in dedicated regions of memory.

82. As to point (11) the examiner disagrees because Maitra memory 420 can store instructions or code that are part of application programs (e.g. col. 8, lines 1-4). Since these applications programs can run at different clock cycles, hence different speeds, they would be stored in their own dedicated memory regions.

83. In the remarks, applicants argued that (12) Maitra does not say anything about a software package's stack residing in the software package's memory region.

84. As to point (12) the examiner disagrees because Maitra discusses Application characterization unit 422 comprises the second plurality of processor executable instructions which is executed by 410 processor in the manner shown in FIG. 6. Clock programming unit 423 comprises the third plurality of processor executable instruction stored in memory which is executed by 410 processor (e.g. col. 8, lines 13-19). Each application has its own stack residing within its own application because applications are running at different speeds.

85. In the remarks, applicants argued that (13) Maitra says nothing about performing tests to determine the validity of applications to be executed, and more specifically, says nothing about a test of validity being "whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time.

86. As to point (13) the examiner disagrees because Maitra as part of the application characterization unit discusses the validity of the stack and or heap memory. The task switch detection unit 310 (e.g. col. 7, lines 19-26) determines the applications scheduled

by the operating system to be run by microprocessor during each time quantum by accessing information from task scheduling unit.

87. In the remarks, applicants argued that (14) Maitra does not allow the execution of an application to extend into another application's assigned "time slice". Moreover this is not the limitation of claims 21 and 46.

88. As to point (14) the examiner disagrees because Maitra discusses priorities that can be assigned internally and externally to time slices, once a priority changes one application can be executed in another applications time slice (e.g. col. 4, lines 40-55).

89. In the remarks, applicants argued that (15) Maitra says nothing about detecting the presence of applications to be run.

90. As to point (15) the examiner disagrees because Maitra clearly shows that the task scheduling unit 110 keeps a list of applications (e.g. col. 4, line 38). Keeping a list of applications shows whether an application is present or not.

91. In the remarks, applicants argued that (16) Jablon says nothing about applying checksums in general to a software package's program memory and Jablon does not say anything about the use of one's complement checksums in testing the validity of a software package.

92. As to point (16) the examiner disagrees because Jablon shows the use of checksums in BIOS extensions that are programs that are loaded into memory for execution. The argument the Jablon makes for not using the checksums are in relation to viruses, which is not the subject of applicant's application (e.g. col. 5, lines 4-18).

93. In the remarks, applicants argued that (17) since neither Maitra nor Jablon disclose the limitations of claims 4 and 29, there is obviously no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

94. As to point (17) the examiner disagrees because in response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to test the validity of the software application using a one's compliment checksum method.

95. In the remarks, applicants argued that (18) Nothing is said about one of the tests for validity being whether the address returned for the software package's initialization procedure lies within its dedicated memory region.

96. As to point (18) the examiner disagrees because Jablon discusses the method to allow the memory address space to be portioned and allow control over which software has access to individual regions of the memory. The Jablon method allows software control in one direction: from unprotected to protected mode. Neither claim 4 nor claim

29 specifies in which direction the address is returned; therefore the address control scheme can be interpreted as one directional (e.g. col. 6, lines 35-39).

97. In the remarks, applicants argued that (19) neither Maitra nor Jablon discloses the limitations of claims 4 and 29, there is obviously no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

98. As to point (19) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to have set memory partitions for each software package and the validity test being conducted within predefined memory regions.

99. In the remarks, applicants argued that (20) nothing is said about safety critical software being placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

100. As to point (20) the examiner disagrees because Jablon discusses methods for trusted software to both enable and disable the protection mechanism for a given region

of memory (e.g. col. 6, lines 27-38). Jablon discusses a protection mechanism for code to be places in a protectable non-volatile storage area (e.g. Abstract).

101. In the remarks, applicants argued that (21) since neither Maitra nor Jablon disclose the limitations of claims 18 and 43, there is obviously no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

102. As to point (21) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to isolate different software packages in different partitions.

103. In the remarks, applicants argued that (22) the passage in Jablon is not a disclosure of software package being enableable to read data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks readable by the software packages in accordance with a set of one or more rules.

104. As to point (22) the examiner disagrees because the latch system discussed by Jablon and its associated rules allow software programs to access memory blocks for reading and writing during the startup procedure (e.g. col. 8, lines 20-34).

105. In the remarks, applicants argued that (23) since neither Maitra not Jablon disclose limitations of claims 19 and 44, there is obviously no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

106. As to point (23) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to provide for software packages to read and write memory blocks from other software packages in accordance with a predetermined set of rules.

107. In the remarks, applicants argued that (24) the passage in Jablon is not a disclosure of a software package being enableable to write data only from zero or more memory blocks associated with other software packages, the zero or more memory blocks writeable by a software package being either predetermined or determined

during execution of the software packages in accordance with a set of one or more rules.

108. As to point (24) the examiner disagrees because the latch system discussed by Jablon and its associated rules allow software programs to access memory blocks for reading and writing during the startup procedure (e.g. col. 8, lines 20-34).

109. In the remarks, applicants argued that (25) since neither Maitra not Jablon disclose limitations of claims 20 and 45, there is obviously no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

110. As to point (25) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Jablon in order to provide for software packages to read and write memory blocks from other software packages in accordance with a predetermined set of rules.

111. In the remarks, applicants argued that (26) neither the cooperative multitasking operating system nor the time-slice multitasking operating system discloses applicants'

limitation "the background tasks being performed after the foreground tasks have been completed".

112. As to point (26) the examiner disagrees because Reznak teaches about a mechanism whereby processing time is divided between the background and foreground tasks. Moreover, multitasking operating systems include scheduling facilities that utilize processing time estimates provided by tasks prior to dispatch to allocate processing resources to the tasks (e.g. col. 1, lines 25-32 and lines 51-56).

113. In the remarks, applicants argued that (27) neither Maitra nor Reznak disclose the limitations of claims 10 and 35, there is no motivation for a person skilled in the art for incorporating such limitations in the Maitra invention.

114. As to point (27) the examiner disagrees In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Reznak in order to have a background and a foreground application processing whereby the background process executes when the foreground processes cease to operate.

115. In the remarks, applicants argued that (28) Maitra says nothing about the power utilized in executing the software package being minimized after completion of the background tasks.

116. As to point (28) the examiner disagrees because, Maitra teaches the method wherein the software package causes the power utilized in executing the software package to be minimized after completion of the background task (e.g. col. 2, lines 39-48).

117. In the remarks, applicants argued that (29) the infinite loop of Davidson et al. is not a background task.

118. As to point (28) the examiner disagrees because Davidson teaches the method wherein a background task is an infinite loop (e.g. col. 17, lines 32-58).

119. In the remarks, applicants argued that (30) The combination of Maitra, Reznak and Davidson does not disclose applicants claim limitations and there is no motivation for providing such limitations in Maitra invention.

120. As to point (30) the examiner disagrees In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at

the time the invention was made to combine Maitra and Davidson in order to have the background process run in an infinite loop.

121. In the remarks, applicants argued that (31) failure in a startup of a software package is not a disclosure of a failure to execute the software package.

122. As to point (31) the examiner disagrees because Yen teaches a method wherein a failure in the execution of a software package causes information to be logged in a failure log (e.g. col. 6, lines 49-59).

123. In the remarks, applicants argued that (32) the combination of Maitra and Yen does not disclose applicants claim limitation and there is no motivation for providing such a limitation in Maitra invention.

124. As to point (32) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Yen in order to log the failure of a software package.

125. In the remarks, applicants argued that (33) there is no disclosure pertaining to Multi-tasking computer systems and failures in execution of one software package caused by another software package.

126. As to point (33) the examiner disagrees because Yen shows a method wherein a failure in execution is linked to the software package that caused the failure (e.g. col. 5, lines 1-16 and Figures 3 and 9).

127. In the remarks, applicants argue that (34) the combination of Maitra and Yen does not disclose applicants claim limitation and there is no motivation for providing such a limitation in the Maitra invention.

128. As to point (34) the examiner disagrees because In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Yen in order to observe whether or not the failure is caused by a software package.

129. In the remarks, applicants argue that (35) nothing is disclosed concerning quality of performance. Nothing is disclosed concerning performance quality parameters. Nothing is disclosed concerning the determination of the values of the performance

quality parameters from information logged in a failure log. Nothing is disclosed concerning the execution of a software package being subject to a plurality of execution options. And nothing disclosed concerning an execution option being selected on the basis of one or more performance-quality parameter values.

130. As to point (35) the examiner disagrees because Harper teaches that DAC unit receives performance parameters either from the operating system or from a log file and send these parameters to the XTALK module. The chain of resources is a simple circular linked list that contains description of the parameters that are to be monitored for resource exhaustion. Figures 16 and 17 show some of the performance parameters (e.g. col. 13, lines 35-67 and col. 14, lines 1-35).

131. In the remarks, applicants argue that (36), the combination of Maitra and Harper does not disclose applicants claim limitations and there is no motivation for providing such limitations in the Maitra invention.

132. As to point (36) the examiner disagrees In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Harper in order to provide for a

performance based system whereby the log file can monitor and determine the packages performance.

133. In the remarks, applicants argue that (37) the passage in Potter does not disclose a method for repetitively executing a plurality of software packages (claim 1) wherein one or more of the plurality of software packages are independently compiled, linked and loaded.

134. As to point (36) the examiner disagrees because Potter discusses application 11 along with the run time library that is written in high level languages are dynamically compiled linked and loaded (e.g. col. 7, lines 40-57).

135. In the remarks, applicants argue (38) that the combination of Maitra and Potter does not disclose applicants claim limitation and there is no motivation for providing such a limitation in the Maitra invention.

136. As to point (38) the examiner disagrees In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Potter to provide for a software package to be independently complied, linked and loaded by the Operating System.

137. In the remarks, applicants argue (39) that the mere mention of "user programs" and "stack" in the same sentence does not disclose multi-tasking computer structure wherein one or more of the software packages being executed may have their own stacks.

138. As to point (39) the examiner disagrees because Circello shows a method wherein a software package has its own stack, the software package's stack being selected prior to executing the software package (e.g. col. 35-51). Moreover Circello teaches about the user programs routinely manipulate the contents of the stack pointer which defines the top of the stack.

139. In the remarks, applicants argue (40) that the combination of Maitra and Circello does not disclose applicants' claim limitation and there is no motivation for providing such a limitation in the Maitra invention.

140. As to point (40) the examiner disagrees, In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Maitra and Circello in order for the software's stack to be executed and selected by a process.

141. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Farhood Moslehi whose telephone number is 703-305-8646. The examiner can normally be reached on M-F 8:30-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee can be reached on 703-305-8498. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

fm

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100